Personalised Human-Robot Co-Adaptation in Instructional Settings using Reinforcement Learning

Yuan Gao[1], Wolmet Barendregt[2], and Ginevra Castellano[1]

[1]Department of Information Technology, Uppsala University, Uppsala, Sweden
[2]Department of Applied IT, Gothenburg University, Gothenburg, Sweden

Abstract. In the domain of robotic tutors, personalised tutoring has started to receive scientists' attention, but is still relatively underexplored. Previous work using reinforcement learning (RL) has addressed personalised tutoring from the perspective of affective policy learning. In this paper we build on previous work on affective policy learning that used RL to learn what robot's supportive behaviours are preferred by users in an educational scenario. We propose a RL framework for personalisation that selects a robot's supportive behaviours to maximize user's task performance in a learning scenario where a Pepper robot acting as a tutor helps people learning how to solve grid-based logic puzzles. This work is relevant for the development of persuasive embodied agents and social robots used to support users in different scenarios. In particular, this paper makes a contribution towards the development of algorithms for human-robot co-adaptation that enable robots and agents to select effective strategies to establish long-term relationships with human users.

Keywords: Social robots, Reinforcement learning, Personalisation, Tutoring systems

1 Introduction

Robots are now used to support humans in new social roles, such as providing assistance for the elderly at home, serving as tutors, acting as therapeutic tools for children with autism, or as game companions for entertainment purposes [1]. However, human social skills remain unmatched in robots. To meet the demands of Europe's citizens in the 21st century, our prospective robotic companions need to learn to interact socially with humans [5] and adapt to their needs, preferences, interests, and emotions in order to become highly personalised to their users. Simulating the tremendous social adaptation abilities that characterise human interactions requires the establishment of bidirectional processes in which humans and robots synchronise and adapt to each other in real-time by means of an exchange of verbal and non-verbal behaviours (e.g., facial expressions, gestures, speech) in order to achieve mutual co-adaptation.

In recent years, technical advances in machine learning methods [9] have opened the door to new ways of building co-adaptive human-robot interactive systems. In the domain of robotic tutors [3], which are used to support student learning in educational scenarios, personalised tutoring has started to receive scientists' attention, but is still relatively underexplored, especially when it comes to build robot abilities enabling robots to interact and adapt to users over extended periods of time.

In the social human-robot interaction (HRI) literature, personalised tutoring has started to be addressed from the perspective of affective policy learning: affect and affect-related states such as engagement have been used to build reward signals in reinforcement learning (RL)-based frameworks to select motivational strategies [6] or supportive behaviours [8] personalised to each student. RL-based approaches have also been proposed to decide how to employ different social behaviors to achieve interactional goals in task-oriented HRI [7]. Moreover, dynamic probabilistic models and Bayesian networks have been used in robotic tutors to model learner's skills and behaviours and their relationships with a robot's tutoring actions [12] and to assess learner's skills to deliver personalised lessons [10].

We build on previous work on affective policy learning that used RL to learn what robot's supportive behaviours are preferred by users in an educational scenario [8]. In this work we take a step forward and propose a RL framework for personalisation that selects a robot's supportive behaviours to maximize user's task performance in a learning scenario where a Pepper [1] robot acting as a tutor helps people learning how to solve grid-based logic puzzles. This work is relevant for the development of persuasive embodied agents and social robots used to support users in different scenarios. In particular, this paper makes a contribution towards the development of algorithms for human-robot co-adaptation that enable robots and agents to select effective strategies to establish long-term relationships with human users.

2 Scenario

We developed a scenario where a Pepper robot acting as a robotic tutor helps people solve grid-based logic puzzles called nonograms. These have previously been used to study robot personalisation to people's learning differences [10]. Nonograms have the advantage of not being well known to most people, thus ensuring that users interacting with the robot start the task-oriented interaction with the robot with a similar skill level.

In the task, the user is asked to solve several nonogram puzzles while a Pepper robot stands in front of the user observing their progress, learning a user model based on the interaction process, and generating verbal utterances in order to provide social support to the user during learning. Robot personalisation to individual users is achieved by combining a decision tree model with a Multi-Armed Bandit (MAB) algorithm called Exponential-Weight Algorithm for Exploration and Exploitation (Exp3) [2] to learn which robot's sup-

[1] https://www.ald.softbankrobotics.com/en/cool-robots/pepper

portive behaviors (described in Section 3.2) increase users' task performance in the puzzle-solving task.

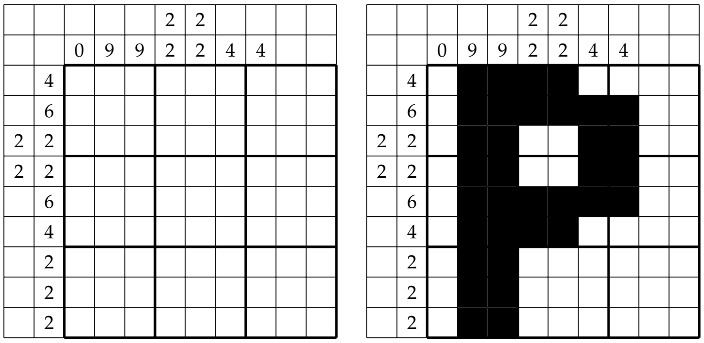
2.1 Nonograms

Nonograms are puzzles where cells in a grid must be filled with black or left blank. In these puzzles, the numbers indicate how many black lines are needed to fill continuous lines for each row or column.

Formally, there are three rules to be followed in this game:

1. Each cell must be colored (black) or left empty (white).
2. If a row or column has k numbers: s1,s2,...,sk , then it must contain k black runsthe first (leftmost for rows/topmost for columns) black run with length s1, the second black run with length s2, and so on.
3. There should be at least one empty cell between two consecutive black runs.

By following these three rules, the participant should find a solution specified by the numbers around the grid.



Unsolved nonogram

Solved nonogram

Fig. 1. An unsolved nonogram puzzle (left) and its corresponding solution (right).

3 System

The system consists of different components: the nonogram interface, the user model, and the personalisation module, which consists of the Exp3 module and an action selection module. Figure 2 shows the relationships between the different components. The robot monitors the user's progress in the task through the nonogram interface and builds a user model extracting task indicators that convey information about whether the user is experiencing difficulties during the puzzle solving task. If that is the case, the personalisation module selects a category of supportive behaviour based on a policy learned by the Exp3 algorithm. The selected category will then be passed to an action selection module using a decision tree, which will choose the most relevant robot action for the current situation according to the selected category. In the following sections, we describe the different components of the system.

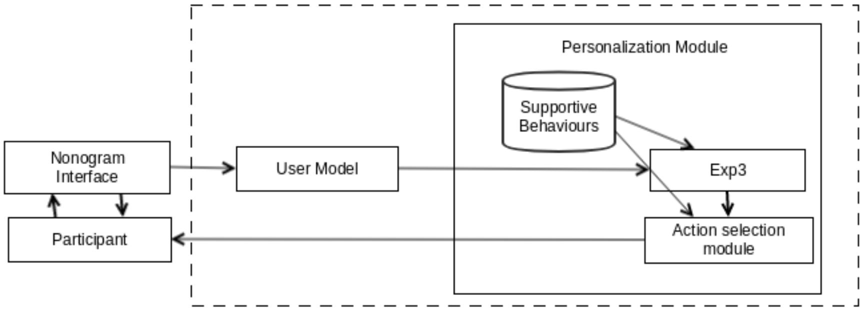


Fig. 2. System components. Arrows indicate the flow of information. The dotted line means that User Model and Personalisation Module are integrated together as one software system.

3.1 User model

The user model extracts a number of tasks indicators:

- **TimeLastMove** It measures the time taken to make the last action.
- **TimeLastSetOfMoves** It measures the time taken to make the last N actions, where N is a pre-defined arbitrary number.
- **CorrectMove** It measures whether the last action made by the user is correct or not.

These are used to define a set of rules that assess whether the user is experiencing difficulties in the puzzle solving task. The rules take the user's actions and their corresponding time into consideration. For example, if an action takes the user more than T seconds to complete, where T is an arbitrary number, then it may indicate that the user experienced difficulty in the last decision.

The user model combines all the information it gathers from the task indicators to make a final decision whether the robot should generate an action or not. This decision is then passed to the personalisation module.

3.2 Personalisation module

The personalisation module uses a RL-based approach that learns which supportive behaviours delivered by the robot maximize user task performance. The latter is defined as the time taken to the user to complete the nonogram puzzle. This is a problem of policy learning, which in a RL framework means optimising action selection policies to maximise a reward. The general idea here is for the robot to learn a policy of optimal supportive behaviours that maximise a user's task performance. Building on previous work on affect co-adaptation mechanisms for a social robot [8], we model this problem as a MAB problem and use an algorithm from the set of MAB learning algorithms – Exp3 [2].

Supportive behaviours We design robot behaviours in the form of verbal utterances by adopting the categorization proposed by Cutrona [4]. We select supportive behaviours belonging to four different categories, namely information support, tangible support, esteem support and emotional support. It has been shown that people differ in their preference for social support [11]. In the following table, we give example of different categories of supportive behaviours.

Information support	”Do you need more information about the rules?”
Tangible support	”If you feel it is difficult, I can help you by completing the next one.”
Esteem support	”The game is hard this time.”
Emotional support	”But please don't worry, I am here for you.”

Table 1. Examples of supportive behaviours implemented in the Pepper robot .

Exp3 The system is modelled as an optimization process of different supportive behaviours under a framework of multi-armed bandit problem. In our case, generating appropriate behaviours for different participants is the goal of the algorithm. In the following text, we explain how does the Exp3 algorithm optimize the probability distribution over all categories of supportive behaviours.

We connect different category of supportive behaviours with different actions in Exp3. Considering a process with K different actions, the Exp3 algorithm [2] functions as described in Algorithm 1, where γ is the exploration factor, and w_i is the weight of each action i. $p_i(t)$ is the probability of selecting action i at round t, while T means the total number of iterations. At the beginning, the algorithm initializes the exploration parameter γ. This parameter adjusts the possibility that the algorithm attempts to execute other actions while a certain action already has the highest probability. Next, the algorithm associates a weight with each action in order to give each action a probability to form a distribution over all possible actions.

After the exploration, the algorithm iterates T times the learning procedure, in order to learn from the environment and to generate a better probability distribution to receive more accumulative reward from the environment. In the learning procedure, the algorithm selects an action i based on the distribution $\mathcal{P}$, and then receives a reward $x_{i_t}(t)$ from the environment. Thereafter, an estimated reward $\hat{x}_{i_t}(t)$ is calculated as $x_{i_t}(t)/p_{i_t}(t)$ to further include the influence of the probability on the reward. In the end, the algorithm updates the weight associated with the action, while the weights of other actions ($w_j, \forall j \neq i_t, j \in \{1, \dots, K\}$) remain the same. After the algorithm converges, the eventual probability distribution over different actions is considered to be the best(and sometimes final) strategy of maximizing the reward.

Algorithm 1 Exp3

```
procedure INITIALIZATION
    initialize $\gamma \in [0,1]$
    initialize $w_i(1) = 1, \forall i \in \{1, \dots, K\}$
    for distribution $\mathcal{P}$,
        set $p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}, \forall i \in \{1, \dots, K\}$
end procedure
procedure ITERATION
    repeat
        draw $i_t$ according to $\mathcal{P}$
        observe reward $x_{i_t}(t)$
        define the estimated reward $\hat{x}_{i_t}(t)$ to be $x_{i_t}(t)/p_{i_t}(t)$
        set $w_{i_t}(t+1) = w_{i_t}(t)e^{\gamma \hat{x}_{i_t}(t)/K}$
        set $w_j(t+1) = w_j(t), \forall j \neq i_t \; and \; j \in \{1, \dots, K\}$
        update $\mathcal{P}$:
            $p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}, \forall i \in \{1, \dots, K\}$
    until $T$ times
end procedure
```

To well integrate Exp3 in our system, each action in this algorithm is associated with a possible category of supportive behaviours, which are described in Section 3.2. In each iteration, the probability of selecting a certain action is adapted to the current environment. For instance, there are four actions ($K = 4$) in the learning procedure of algorithm by design, i.e., action 1, 2, 3, and 4. Respectively, actions 1, 2, 3 and 4 are mapped to four different categories of supportive behaviours: the robot can choose to select information support, tangible assistance, esteem support or emotional support.

That is, if the randomly sampled category of supportive behaviours i is 1, then the robot decides to use information support. After the algorithm receives the feedback, the weight of the corresponding action (i.e., action 1) is updated based on:

$$w_{1_t}(t+1) = w_{1_t}(t)e^{\gamma \hat{x}_{1_t}(t)/4}. \tag{1}$$

The weights of other actions (i.e., action 2 3, and 4) stay the same. In the final step, the distribution $\mathcal{P}$ is renewed to prepare for the next iteration round according to the following formula:

$$p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{4} w_j(t)} + \frac{\gamma}{4}, \forall i \in \{1,2,3,4\}. \tag{2}$$

Until then, one learning iteration is done. The iteration continues in total T times. By design, the system's default T value is set to 200 $T = 200$, which means a fixed learning period for 200 iterations.

Action selection module After a category of supportive behaviours is selected by the Exp3 algorithm, the action selection module, which consists of a decision tree, checks if the selected category is appropriate to the current situation. If the latter is not appropriate (for example, when user has played several games and makes a mistake at the beginning of a new game, the information support of explaining the rules of the game is not necessary, it is more likely that the user simply made a mistake), the robot will not perform any action, otherwise the robot will choose a specific robot's action (i.e., supportive behaviour) from a pool of available actions within that category.

4 Methodology

4.1 Experimental set up

As illustrated in Figure 3, the experimental setting includes a Pepper robot, a 27 inches IIYAMA touch screen placed on a table, an ubuntu 14.04 Linux server, a Microsoft Kinect V2 camera, a Logitech C920 1080p webcam and a laptop. The user is sitting on a chair in front of the robot, with the touch screen and the table placed between the user and the robot.

The Kinect V2 camera, placed in front of the user, is used for recording user behavioural data; the Logitech webcam, positioned on a tripod on the side of the table, is connected to the laptop to record videos for offline video analysis. The software runs on a Linux server and consists two parts. One part contains the nonogram interface that interacts with the user and an algorithmic module that updates the parameters for the user model. The other part takes the responsibility of controlling the robot and generation of verbal utterances.

4.2 Planned experiment

We plan an experiment with a between-group design (Figure 4, where participants will be randomly assigned to two different conditions, corresponding to two different parameterisations of the robot's behaviour: (1) Personalised condition, where the MAB-based personalisation module will be used, vs Non-personalised condition, where the robot's supportive behaviours will be randomly selected.

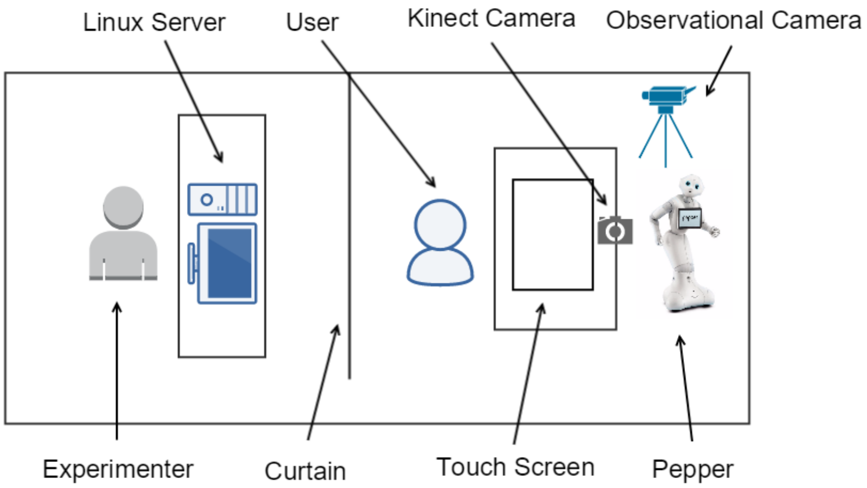


Fig. 3. An illustration of the experimental set up.

During the experiment, each group will go through three sessions, namely a pre-interaction session s_p, a human-robot interaction session s_i and an after-interaction session s_a. In the pre-interaction session, the participants are asked to solve N_p nonogram puzzles (difficulty level l) on their own. In the human-robot interaction session, the participants are asked to solve N_i nonogram puzzles with the assistance of a robot. In the after-interaction session, the participants are asked to solve N_a nonogram puzzles on their own (difficulty level l).

Normally, due learning effects, the average time taken to solve nonogram puzzles in the after-interaction session is shorter than the average time taken to solve nonogram puzzles in the pre-interaction session. Here, we hypothesize that personalisation will affect the time taken by people to solve the puzzle, i.e., with the personalised robot, the difference in the time taken by people to solve the puzzle between pre- and after- interaction session will be more obvious.

Procedure The experiment will take place in a laboratory at Uppsala University and will involve a fully autonomous robot interacting with the two groups of participants (20 for each group). Figure 3 shows the arrangement of the experimental area. A curtain separates the experimental area from the researcher running the experiment. The researcher can observe the experiment via webcam.

Before the experiment starts, the participant will be given a document that describes the experiment and tasks that they need to solve and a consent form

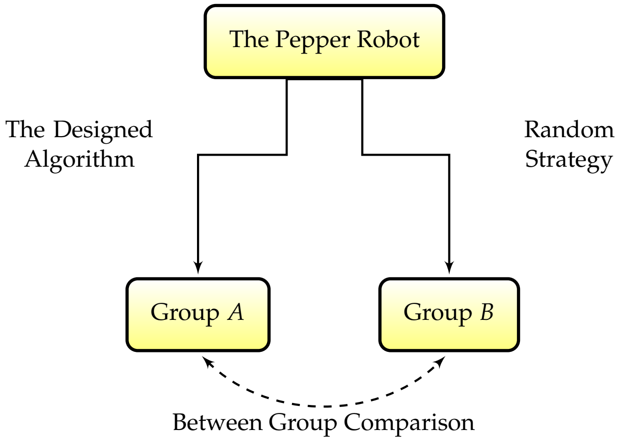


Fig. 4. The Pepper robot applies two different strategies to two groups of people. The only difference between group A and group B is that group A has the designed personalisation algorithm implemented in the human-robot interaction session, whilst group B only a random strategy.

that they need to sign. After signing the consent form, the participant will be asked to enter the experiment area.

The experiment will start with pre-interaction session s_p. In this session, the participant will be asked to complete one middle level nonogram. After the participant completes pre-interaction session s_p, the robot Pepper will be activated and the participant will be asked to solve seven nonogram puzzles with the help of the robot. We chose to include seven nonograms in the interactive session because during preliminary tests of the system we found that normally five to seven nonograms are necessary for the learning process to converge.

Then the robot will be deactivated and the participant will be asked to solve another nonogram in the after-interaction session s_a. After the after-interaction session, the participant will be asked to walk out of the experimental area. In the end, a researcher will conduct a very short interview and ask participants to fill in a set of questionnaires to collect information on the user experience and perception of the robot.

4.3 Preliminary results and conclusion

To date the system has been tested with five participants who took part in a set of pilot studies. Initial results indicate that the system personalised to people over the duration of the study and it did so in a different manner (i.e. converging on the selection of a specific supportive behaviour) for different participants. In the future, we will conduct the full experiment to test our main hypothesis.

5 Acknowledgements

This work was partly supported by the Swedish Research Council (grant n. 2015-04378) and the COIN project (RIT15-0133) funded by the Swedish Foundation for Strategic Research.

References

1. Breazeal, C.L.: Designing sociable robots. MIT press (2004)
2. Bubeck, S., Cesa-Bianchi, N., et al.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends® in Machine Learning 5(1), 1–122 (2012)
3. Castellano, G., Paiva, A., Kappas, A., Aylett, R., Hastie, H., Barendregt, W., Nabais, F., Bull, S.: Towards empathic virtual and robotic tutors. In: International Conference on Artificial Intelligence in Education. pp. 733–736. Springer (2013)
4. Cutrona, C.E., Suhr, J.A., MacFarlane, R.: Interpersonal transactions and the psychological sense of support. Personal relationships and social support pp. 30–45 (1990)
5. Dautenhahn, K.: Socially intelligent robots: dimensions of human–robot interaction. Philosophical Transactions of the Royal Society of London B: Biological Sciences 362(1480), 679–704 (2007)
6. Gordon, G., Spaulding, S., Westlund, J.K., Lee, J.J., Plummer, L., Martinez, M., Das, M., Breazeal, C.: Affective personalization of a social robot tutor for children's second language skills. In: AAAI. pp. 3951–3957 (2016)
7. Hemminghaus, J., Kopp, S.: Towards adaptive social behavior generation for assistive robots using reinforcement learning. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. pp. 332–340. ACM (2017)
8. Leite, I., Pereira, A., Castellano, G., Mascarenhas, S., Martinho, C., Paiva, A.: Social robots in learning environments: a case study of an empathic chess companion. In: Proceedings of the International Workshop on Personalization Approaches in Learning Environments. vol. 732, pp. 8–12 (2011)
9. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. Journal of Machine Learning Research 17(39), 1–40 (2016)
10. Leyzberg, D., Spaulding, S., Scassellati, B.: Personalizing robot tutors to individuals' learning differences. In: Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction. pp. 423–430. ACM (2014)
11. Reevy, G.M., Maslach, C.: Use of social support: Gender and personality differences. Sex roles 44(7), 437–459 (2001)
12. Schodde, T., Bergmann, K., Kopp, S.: Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making. Proceedings of ACM/IEEE HRI 2017 (2017)